

# Control refinement for discrete-time descriptor systems: a behavioural approach via simulation relations

F. Chen<sup>\*</sup>, S. Haesaert<sup>\*</sup>, A. Abate<sup>\*\*</sup>, and S. Weiland<sup>\*</sup>

<sup>\*</sup> *Department of Electrical Engineering  
Eindhoven University of Technology, Eindhoven, The Netherlands*

<sup>\*\*</sup> *Department of Computer Science  
University of Oxford, Oxford, United Kingdom*

---

**Abstract:** The analysis of industrial processes, modelled as descriptor systems, is often computationally hard due to the presence of both algebraic couplings and difference equations of high order. In this paper, we introduce a control refinement notion for these descriptor systems that enables analysis and control design over related reduced-order systems. Utilising the behavioural framework, we extend upon the standard hierarchical control refinement for ordinary systems and allow for algebraic couplings inherent to descriptor systems.

---

*Keywords:* Descriptor systems, simulation relations, control refinement, behavioural theory.

---

## 1. INTRODUCTION

Complex industrial processes generally contain algebraic couplings in addition to differential (or difference) equations of high order. These systems, referred to as descriptor systems (Kunkel and Mehrmann, 2006; Dai, 1989), are commonly used in the modelling of mechanical systems. The presence of algebraic equations, or couplings, together with large state dimensions renders numerical simulation and controller design challenging. Instead model reduction methods (Antoulas, 2005) can be applied to replace the systems with reduced order ones. Even though most methods have been developed for systems with only ordinary difference equations, recent research also targets descriptor systems (Cao et al., 2015).

In this paper, we newly target the use of descriptor systems of reduced order for the verifiable design of controllers. A rich body of literature on verification and formal controller synthesis exists for systems solely composed of difference equations. This includes the algorithmic design of certifiable (hybrid) controllers and the verification of pre-specified requirements (Tabuada, 2009; Kloetzer and Belta, 2008). Usually, these methods first reduce the original, concrete systems to abstract systems with finite or smaller dimensional state spaces over which the verification or controller synthesis can be run. A such controller obtained for the abstract system can be refined over the concrete system leveraging the existence of a similarity relation, e.g., an (approximate) simulation relation, between the two systems (Tabuada, 2009; Girard and Pappas, 2011). For the application of these relations in control problems, a hierarchical control framework is presented by (Girard and Pappas, 2009). Currently, the control synthesis over descriptor systems cannot be dealt with in this fashion due to the presence of algebraic equations.

The presence of similarity relations between descriptor systems has also been a topic under investigation

in (Megawati and Van der Schaft, 2015). This work on similarity relations deals with continuous-time descriptor systems that are unconstrained and non-deterministic, and focuses on the conditions for bisimilarity and on the construction of similarity relations. Instead in this work, we specifically consider the control refinement problem for discrete-time descriptor systems via simulation relations within a behavioural framework, such that properties verified over the future behaviour of the abstract system are also verified over the concrete controlled system. Within the behavioural theory (Willems and Polderman, 2013), a formal distinction is made between a system (its behaviour) and its representations, enabling us to investigate descriptor systems and refinement control problems without having to directly deal with their inherent anti-causality.

In the next section, we define the notion of dynamical systems and control within a behavioural framework and use it to formalise the control refinement problem. Subsequently, Section 3 is dedicated to the exact control refinement for descriptor systems and contains the main results of the paper. The last section closes with the conclusions.

## 2. THE BEHAVIOURAL FRAMEWORK

### 2.1 Discrete-time descriptor systems

As introduced by (Willems and Polderman, 2013), we define dynamical systems as follows.

*Definition 1.* A dynamical system  $\Sigma$  is defined as a triple  $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$

with the *time axis*  $\mathbb{T}$ , the *signal space*  $\mathbb{W}$ , and the *behaviour*  $\mathfrak{B} \subset \mathbb{W}^{\mathbb{T}}$ .  $\square$

In this definition,  $\mathbb{W}^{\mathbb{T}}$  denotes the collection of all time-dependent functions  $w : \mathbb{T} \rightarrow \mathbb{W}$ . The set of trajectories or time-dependent functions given by  $\mathfrak{B}$  represents the

trajectories that are compatible with the system. This set is referred to as the behaviour of the system (Willems and Polderman, 2013). Generally, the representation of the behaviour of a dynamical system by equations, such as a set of ordinary differential equations, state space equations and transfer functions, is non-unique. Hence we distinguish a dynamical system (its behaviour) from the mathematical equations used to represent its governing laws.

We consider dynamical systems evolving over discrete-time ( $\mathbb{T} := \mathbb{N} = \{0, 1, 2, \dots\}$ ) that can be represented by a combination of linear difference and algebraic equations. The dynamics of such a linear discrete-time descriptor system (DS) are defined by the tuple  $(E, A, B, C)$  as

$$\begin{aligned} Ex(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \quad (1)$$

with the state  $x(t) \in \mathbb{X} = \mathbb{R}^n$ , the input  $u(t) \in \mathbb{U} = \mathbb{R}^p$ , and the output  $y(t) \in \mathbb{Y} = \mathbb{R}^k$  and  $t \in \mathbb{N}$ . Further,  $E, A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$  and  $C \in \mathbb{R}^{k \times n}$  are constant matrices and we presume that  $\text{rank}(B) = p$  and  $\text{rank}(C) = k$ .

We say that a trajectory  $w = (u, x, y)$ , with  $w : \mathbb{N} \rightarrow (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$ , satisfies (1) if for all  $t \in \mathbb{N}$  the equations in (1) evaluated at  $u(t), x(t), x(t+1), y(t)$  hold. Then the collection of all trajectories  $w$  defines the *full* behaviour, or equivalently the *input-state-output* behaviour as

$$\mathfrak{B}_{i/s/o} := \{(u, x, y) \in (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})^{\mathbb{N}} \mid (1) \text{ is satisfied}\}. \quad (2)$$

The variable  $x$  is considered as a latent variable, therefore the *manifest*, or equivalently the *input-output* behaviour associated with (1) is defined by

$$\mathfrak{B}_{i/o} := \{(u, y) \mid \exists x \in \mathbb{X}^{\mathbb{N}} \text{ s.t. } (u, x, y) \in \mathfrak{B}_{i/s/o}\}.$$

If  $E$  is non-singular, we refer to the corresponding dynamical system as a *non-singular DS*. In that case, we can transform (1) into standard state space equations, as

$$\begin{aligned} x(t+1) &= \tilde{A}x(t) + \tilde{B}u(t), \\ y(t) &= Cx(t), \end{aligned} \quad (3)$$

with  $\tilde{A} = E^{-1}A$ ,  $\tilde{B} = E^{-1}B$ . Further  $\mathfrak{B}_{i/s/o}$  as in (2) is

$$\{(u, x, y) \in (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})^{\mathbb{N}} \mid (u, x, y) \text{ s.t. (3) holds}\}.$$

Similarly, if  $E$  is non-singular,  $\mathfrak{B}_{i/o}$  can be defined by (3).

The tuple with dynamics (1) defines a dynamical system  $\Sigma$  evolving over the combined signal space  $\mathbb{W} = \mathbb{U} \times \mathbb{X} \times \mathbb{Y}$  with behaviour  $\mathfrak{B} := \mathfrak{B}_{i/s/o}$  given in (2). Similarly, for  $\mathbb{W}$  restricted to input-output space, the tuple  $(\mathbb{N}, \mathbb{U} \times \mathbb{Y}, \mathfrak{B}_{i/o})$  defines the manifest or induced dynamical system.

We are specifically interested in the behaviour initialised at  $t = 0$  with a given set of initial states  $\mathbb{X}_0 \subset \mathbb{X}$ . For this, we say that a trajectory  $w : \mathbb{N} \rightarrow (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$  is initialised with  $\mathbb{X}_0$  if (1) holds and  $x(0) = x_0 \in \mathbb{X}_0$ . Such a trajectory, initialised with  $x_0 \in \mathbb{X}_0$ , is also called the continuation of  $x_0$ . We refer to the collection of initialised trajectories related to  $\mathbb{X}_0$  as the initialised behaviour  $\mathfrak{B}_{i/s/o}^{init}$ . This allows us to formalise our definition of the descriptor system evolving over  $\mathbb{N}$ .

**Definition 2.** (Discrete-time descriptor systems (DS)). A (discrete-time) descriptor system is defined as a dynamical system  $\Sigma$  initialised with  $\mathbb{X}_0$ , whose behaviour can be represented by the combination of algebraic equations and difference equations given in (1), that is

$$\Sigma := (\mathbb{T}, \mathbb{W}, \mathfrak{B}) = (\mathbb{N}, \mathbb{U} \times \mathbb{X} \times \mathbb{Y}, \mathfrak{B}_{i/s/o}^{init}) \quad (4)$$

with

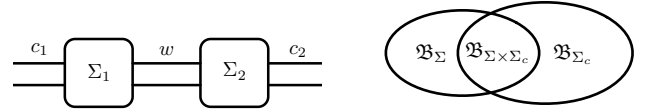
- the time axis  $\mathbb{T} := \mathbb{N} = \{0, 1, 2, \dots\}$ ,
- the full signal space  $\mathbb{W} := \mathbb{U} \times \mathbb{X} \times \mathbb{Y}$ , and
- the *initialised* behaviour<sup>1</sup>

$$\mathfrak{B}_{i/s/o}^{init} = \{w \in \mathbb{W}^{\mathbb{N}} \mid w = (u, x, y) \text{ s.t. (1)}$$

and s.t.  $x(0) = x_0 \in \mathbb{X}_0\}$ .

## 2.2 Control of descriptor systems

Controller synthesis amounts to synthesising a system  $\Sigma_c$ , called a controller, which, after interconnection with  $\Sigma$ , restricts the behaviour  $\mathfrak{B}$  of  $\Sigma$  to desirable (or controlled) trajectories. Thus, in the behavioural framework, control is defined through *interconnections* (or via variable sharing as specified next), rather than based on the causal transmission of signals or information, as in classical system theory. Let  $\Sigma_1 = (\mathbb{T}, \mathbb{C}_1 \times \mathbb{W}, \mathfrak{B}_1)$  and  $\Sigma_2 = (\mathbb{T}, \mathbb{C}_2 \times \mathbb{W}, \mathfrak{B}_2)$  be two dynamical systems. Then, as depicted in Fig. 1a and defined in (Willems and Polderman, 2013), the *interconnection* of  $\Sigma_1$  and  $\Sigma_2$  over  $\mathbb{W}$ , denoted by  $\Sigma = \Sigma_1 \times_w \Sigma_2$  with the shared variable  $w \in \mathbb{W}$ , yields the dynamical system  $\Sigma = (\mathbb{T}, \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{W}, \mathfrak{B})$  with  $\mathfrak{B} = \{(c_1, c_2, w) : \mathbb{T} \rightarrow \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{W} \mid (c_1, w) \in \mathfrak{B}_1, (c_2, w) \in \mathfrak{B}_2\}$ .



(a) The interconnected system  $\Sigma$  obtained via the shared variables  $w$  in  $\mathbb{W}$  between dynamical systems  $\Sigma_1$  and  $\Sigma_2$  with signal spaces  $\mathbb{C}_1 \times \mathbb{W}$  and  $\mathbb{C}_2 \times \mathbb{W}$ . (b) The controlled behaviour obtained via the shared variables  $w$  in  $\mathbb{W}$  between dynamical system  $\Sigma$  and its controller  $\Sigma_c$ . The intersection of the behaviours of the dynamical system  $\Sigma$  and its controller  $\Sigma_c$  is given as  $\mathfrak{B}_{\Sigma \times \Sigma_c} = \mathfrak{B}_{\Sigma} \cap \mathfrak{B}_{\Sigma_c}$ .

Fig. 1. The left figure (a) portrays the general interconnection of two dynamical systems. In figure (b), the more specific case of behavioural intersection for a system and its controller is depicted.

Observe that  $w \in \mathbb{W}^{\mathbb{T}}$  contains the signals shared by both  $\Sigma_1$  and  $\Sigma_2$ , while  $c_1 \in \mathbb{C}_1^{\mathbb{T}}$  only belongs to  $\Sigma_1$  and  $c_2 \in \mathbb{C}_2^{\mathbb{T}}$  only belongs to  $\Sigma_2$ . So, in the interconnected system, the shared variable  $w$  satisfies the laws of both  $\mathfrak{B}_1$  and  $\mathfrak{B}_2$ . Note that it is always possible to trivially extend the signal spaces of  $\Sigma_1$  and  $\Sigma_2$  (and the associated behaviour) such that a full interconnection structure is obtained, that is, such that both  $\mathbb{C}_1$  and  $\mathbb{C}_2$  are empty and the behaviour of the interconnected system is  $\mathfrak{B} = \mathfrak{B}_1 \cap \mathfrak{B}_2$ . Hence, a full interconnection of  $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_{\Sigma})$  and  $\Sigma_c = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_{\Sigma_c})$  is simply  $\Sigma \times_w \Sigma_c = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_{\Sigma} \cap \mathfrak{B}_{\Sigma_c})$ , with the intersection of the behaviours, denoted by  $\mathfrak{B}_{\Sigma \times \Sigma_c}$ , as portrayed in Fig. 1b. That is, interconnection and intersection are equivalent in full interconnections.

Further, we define a well-posed controller  $\Sigma_c$  for  $\Sigma$  as follows.

**Definition 3.** Consider a dynamical system  $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ , with initialised behaviour as defined in (4). We say that a system  $\Sigma_c = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_c)$  is a *well-posed* controller for  $\Sigma$  if the following conditions are satisfied:

<sup>1</sup> In the sequel the indexes *init* and *i/s/o* will be dropped.

- (1)  $\mathfrak{B}_{\Sigma \times \Sigma_c} := \mathfrak{B}_{\Sigma} \cap \mathfrak{B}_{\Sigma_c} \neq \{\emptyset\}$ ;
- (2) For every initial state  $x_0 \in \mathbb{X}_0$ , there exists a unique continuation in  $\mathfrak{B}_{\Sigma \times \Sigma_c}$ .

Denote with  $\mathfrak{C}(\Sigma)$  the collection of all well-posed controllers for  $\Sigma$ .

We want a controller that accepts any initial state of the system. This is formalised in the second condition by requiring that for any initial state of  $\Sigma$ , there exists a unique continuation in  $\mathfrak{B}_{\Sigma \times \Sigma_c}$ . We elucidate the properties of a well-posed linear controller as follows.

*Example 1.* For a system  $\Sigma$  as in (1), consider a controller  $\Sigma_c$ , which is a DS, and has dynamics given as

$$E_c x(t+1) = A_c x(t) + B_c u(t), \quad (5)$$

with  $E_c, A_c \in \mathbb{R}^{n_c \times n}$  and  $B_c \in \mathbb{R}^{n_c \times p}$ . Suppose that the controller shares the variables  $u$  and  $x$  with the system  $\Sigma$ . That is,  $w = (u, x)$ . The interconnected system  $\Sigma \times_w \Sigma_c$  yields the state evolutions of the combined system as

$$\begin{bmatrix} E \\ E_c \end{bmatrix} x(t+1) = \begin{bmatrix} A \\ A_c \end{bmatrix} x(t) + \begin{bmatrix} B \\ B_c \end{bmatrix} u(t), \quad (6)$$

and can be rewritten to

$$\begin{bmatrix} E & -B \\ E_c & -B_c \end{bmatrix} \begin{bmatrix} x(t+1) \\ u(t) \end{bmatrix} = \begin{bmatrix} A \\ A_c \end{bmatrix} x(t). \quad (7)$$

If for any  $x(t) \in \mathbb{X}$ , there exists a pair  $(x(t+1), u(t))$  such that (7) holds, then this implies that for any initial state  $x_0 \in \mathbb{X}_0$  of  $\Sigma$  there exists a continuation in the controlled behaviour. In addition, if the pair  $(x(t+1), u(t))$  is unique for any  $x(t) \in \mathbb{X}$ , then this continuation is unique and we say that  $\Sigma_c \in \mathfrak{C}(\Sigma)$ . This existence and uniqueness of the pairs  $(x(t+1), u(t))$  depends on the solutions of the matrix equality (7). We use the classical results on the solutions of matrix equalities (cf. (Abadir and Magnus, 2005)) to conclude that the first well-posedness condition is satisfied if and only if

$$\text{rank} \left( \begin{bmatrix} E & B \\ E_c & B_c \end{bmatrix} \right) = \text{rank} \left( \begin{bmatrix} E & B & A \\ E_c & B_c & A_c \end{bmatrix} \right). \quad (8)$$

If in addition,

$$\text{rank} \left( \begin{bmatrix} E & B & A \\ E_c & B_c & A_c \end{bmatrix} \right) = n + p, \quad (9)$$

then the second condition is also satisfied and  $\Sigma_c \in \mathfrak{C}(\Sigma)$ .

Of interest is the design of well-posed controllers subject to specifications over the future output behaviour of the controlled system. We thus consider specifications defined over the output space. In order to analyse the output behaviour, we introduce a projection map. For  $\mathfrak{B} \subset (\mathbb{W}_1 \times \mathbb{W}_2)^{\mathbb{T}}$  we denote with  $\Pi_{\mathbb{W}_2}$  a projection given as

$$\Pi_{\mathbb{W}_2}(\mathfrak{B}) := \{w_2 \in \mathbb{W}_2^{\mathbb{T}} | \exists w_1 \in \mathbb{W}_1^{\mathbb{T}} \text{ s.t. } (w_1, w_2) \in \mathfrak{B}\}.$$

We focus here on finding a controller  $\Sigma_c$  for a given dynamical system  $\Sigma$  such that the output behaviour  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c})$  of the interconnected system satisfies some specifications.

### 2.3 Exact control refinement & problem statement

Let us refer to the original DS that represents the real physical system as the *concrete* DS. It is for this system that we would like to develop a well-posed controller. Recall that the DS is a dynamical system  $\Sigma$  with dynamics

$(E, A, B, C)$  as in (1) and initialised with  $\mathbb{X}_0$ . A well-posed controller for  $\Sigma$  is referred to  $\Sigma_c \in \mathfrak{C}(\Sigma)$ . The controlled concrete system is the interconnected system  $\Sigma \times_w \Sigma_c$  with the shared variables  $w = (u, x)$ .

Now, we consider a simpler DS  $\Sigma_a$ , related to the concrete DS  $\Sigma$ , with dynamics given as  $(E_a, A_a, B_a, C_a)$  and initialised with  $\mathbb{X}_{a0}$ . We assume that the synthesis of a well-posed controller  $\Sigma_{c_a}$  for  $\Sigma_a$  is substantially easier than for  $\Sigma$ . We refer to this simpler system  $\Sigma_a$  as the *abstract* DS, and we note that its signals take values  $u_a(t), x_a(t), y_a(t)$  with  $x_a(t) \in \mathbb{X}_a = \mathbb{R}^m, u_a(t) \in \mathbb{U}_a = \mathbb{R}^q, y_a(t) \in \mathbb{Y}_a = \mathbb{Y} = \mathbb{R}^k$  and  $t \in \mathbb{N}$ . With respect to the concrete system, the abstract DS is generally a reduced-order system. The controlled abstract system  $\Sigma_a \times_{w_a} \Sigma_{c_a}$  is the interconnected system with the shared variables  $w_a = (u_a, x_a)$ .

If we assume that we can compute a well-posed controller for the abstract system, then the control synthesis problem reduces to a control refinement problem.

*Definition 4.* (Exact control refinement). Let  $\Sigma_a$  and  $\Sigma$  be the abstract and concrete DS, respectively. We say that controller  $\Sigma_c \in \mathfrak{C}(\Sigma)$  refines the controller  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$  if  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}})$ .

Then we formalise the exact control refinement problem.

*Problem 1.* Let  $\Sigma_a$  and  $\Sigma$  be the abstract and concrete DS, respectively. For any  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$ , refine  $\Sigma_{c_a}$  to  $\Sigma_c$ , s.t.  $\Sigma_c \in \mathfrak{C}(\Sigma)$  and  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}})$ .

In the next section, we will show that the existence of a solution to this problem hinges on certain conditions involving similarity relations between the concrete and abstract DS. For this, we will first introduce simulation relations to formally characterise this similarity.

## 3. EXACT CONTROL REFINEMENT

### 3.1 Similarity relations between DS

We give the notion of simulation relation as defined in (Tabuada, 2009) for transition systems and applied to pairs of DS  $\Sigma_1$  and  $\Sigma_2$  that share the same output space  $\mathbb{Y}_1 = \mathbb{Y}_2 = \mathbb{Y}$ .

*Definition 5.* Let  $\Sigma_1$  and  $\Sigma_2$  be two DS with respective dynamics  $(E_1, A_1, B_1, C_1)$  and  $(E_2, A_2, B_2, C_2)$  over state spaces  $\mathbb{X}_1$  and  $\mathbb{X}_2$ . A relation  $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$  is called a *simulation relation from  $\Sigma_1$  to  $\Sigma_2$* , if  $\forall (x_1, x_2) \in \mathcal{R}$ ,

- (1) for all  $(u_1, x_1^+) \in \mathbb{U}_1 \times \mathbb{X}_1$  subject to  $E_1 x_1^+ = A_1 x_1 + B_1 u_1$  there exists  $(u_2, x_2^+) \in \mathbb{U}_2 \times \mathbb{X}_2$  subject to  $E_2 x_2^+ = A_2 x_2 + B_2 u_2$  such that  $(x_1^+, x_2^+) \in \mathcal{R}$ , and
- (2) we have  $C_1 x_1 = C_2 x_2$ .

We say that  $\Sigma_1$  is simulated by  $\Sigma_2$ , denoted by  $\Sigma_1 \preceq \Sigma_2$ , if there exists a simulation relation  $\mathcal{R}$  from  $\Sigma_1$  to  $\Sigma_2$  and if in addition  $\forall x_{10} \in \mathbb{X}_{10}, \exists x_{20} \in \mathbb{X}_{20}$  such that  $(x_{10}, x_{20}) \in \mathcal{R}$ .

We call  $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$  a *bisimulation relation* between  $\Sigma_1$  and  $\Sigma_2$ , if  $\mathcal{R}$  is a simulation relation from  $\Sigma_1$  to  $\Sigma_2$  and its inverse  $\mathcal{R}^{-1} \subseteq \mathbb{X}_2 \times \mathbb{X}_1$  is a simulation relation from

$\Sigma_2$  to  $\Sigma_1$ . We say that  $\Sigma_1$  and  $\Sigma_2$  are bisimilar, denoted by  $\Sigma_1 \cong \Sigma_2$ , if  $\Sigma_1 \preceq \Sigma_2$  w.r.t.  $\mathcal{R}$  and  $\Sigma_2 \preceq \Sigma_1$  w.r.t.  $\mathcal{R}^{-1}$ .

Simulation relations as defined above are transitive. Let  $\mathcal{R}_{12}$  and  $\mathcal{R}_{23}$  be simulation relations respectively, from  $\Sigma_1$  to  $\Sigma_2$  and from  $\Sigma_2$  to  $\Sigma_3$ . Then a simulation relation from  $\Sigma_1$  to  $\Sigma_3$  is given as a composition of  $\mathcal{R}_{12}$  and  $\mathcal{R}_{23}$ , namely  $\mathcal{R}_{12} \circ \mathcal{R}_{23} = \{(x_1, x_3) \mid \exists x_2 : (x_1, x_2) \in \mathcal{R}_{12} \wedge (x_2, x_3) \in \mathcal{R}_{23}\}$ .

We also have that  $\Sigma_1 \preceq \Sigma_2$  and  $\Sigma_2 \preceq \Sigma_3$  implies  $\Sigma_1 \preceq \Sigma_3$  and, in addition,  $\Sigma_1 \cong \Sigma_2$  and  $\Sigma_2 \cong \Sigma_3$  implies  $\Sigma_1 \cong \Sigma_3$ .

Simulation relations have also implications on the properties of the output behaviours of the two systems. More precisely, if a system is simulated by another system then this implies output behaviour inclusion. This follows from Proposition 4.9 in (Tabuada, 2009) and is formalised next.

**Proposition 6.** Let  $\Sigma_1$  and  $\Sigma_2$  be two DS with simulation relations as defined in Definition 5. Then,

$$\begin{aligned} \Sigma_1 \preceq \Sigma_2 &\implies \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_1}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_2}), \\ \Sigma_1 \cong \Sigma_2 &\implies \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_1}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_2}). \end{aligned}$$

Simulation relations can also be used for the controller design for deterministic systems such as nonsingular DS (Tabuada, 2009; Fainekos et al., 2007; Girard and Pappas, 2009). This will be used in the next subsection, where we consider the exact control refinement for non-singular DS. After that, we introduce a transformation of a singular DS to an auxiliary nonsingular DS representation, referred to as a driving variable (DV) system. The exact control refinement problem is then solved based on the introduced notions.

### 3.2 Control refinement for non-singular DS

Let us consider the simple case where the concrete and abstract systems of interest are given with non-singular dynamics. For these systems, the existence of a simulation relation also implies the existence of an interface function (Girard and Pappas, 2009), which is formulated as follows.

**Definition 7.** (Interface). Let  $\Sigma_1$  and  $\Sigma_2$  be two non-singular DS defined over the same output space  $\mathbb{Y}$  with a simulation relation  $\mathcal{R}$  from  $\Sigma_1$  to  $\Sigma_2$ . A mapping  $\mathcal{F} : \mathbb{U}_1 \times \mathbb{X}_1 \times \mathbb{X}_2 \mapsto \mathbb{U}_2$  is an *interface related to  $\mathcal{R}$* , if  $\forall (x_1, x_2) \in \mathcal{R}$  and for all  $u_1 \in \mathbb{U}_1$ ,  $u_2 := \mathcal{F}(u_1, x_1, x_2) \in \mathbb{U}_2$  is such that  $(x_1^+, x_2^+) \in \mathcal{R}$  with

$$x_1^+ = A_1 x_1 + B_1 u_1 \text{ and } x_2^+ = A_2 x_2 + B_2 u_2.$$

It follows from Definition 5 that there exists at least one interface related to  $\mathcal{R}$  if two deterministic, or non-singular systems are in a simulation relation. As such we can solve the exact refinement problem as follows.

**Theorem 8.** Let  $\Sigma_1$  and  $\Sigma_2$  be two non-singular DS defined over the same output space  $\mathbb{Y}$  with dynamics  $(I, A_1, B_1, C_1)$  and  $(I, A_2, B_2, C_2)$ , which are initialised with  $\mathbb{X}_{10}$  and  $\mathbb{X}_{20}$ , respectively. If there exists a relation  $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$  such that

- (1)  $\mathcal{R}$  is a simulation relation from  $\Sigma_1$  to  $\Sigma_2$ , and
- (2)  $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$  s.t.  $(x_{10}, x_{20}) \in \mathcal{R}$ ,

then for any controller  $\Sigma_{c_1} \in \mathfrak{C}(\Sigma_1)$ , there exists a controller  $\Sigma_{c_2} \in \mathfrak{C}(\Sigma_2)$  that is an exact control refinement for  $\Sigma_{c_1}$  and thus achieves with

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_2 \times \Sigma_{c_2}}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_1 \times \Sigma_{c_1}}).$$

**Proof.** Since  $\mathcal{R}$  is a simulation relation from  $\Sigma_1$  to  $\Sigma_2$ , there exists an interface function  $\mathcal{F} : \mathbb{U}_1 \times \mathbb{X}_1 \times \mathbb{X}_2 \mapsto \mathbb{U}_2$  as given in Definition 7, cf (Tabuada, 2009; Girard and Pappas, 2009). Additionally, due to (2) there exists a map,  $\mathcal{F}_0 : \mathbb{X}_{20} \rightarrow \mathbb{X}_{10}$  such that for all  $x_{20} \in \mathbb{X}_{20}$  it holds that  $(\mathcal{F}_0(x_{20}), x_{20}) \in \mathcal{R}$ .

Next, we construct the controller  $\Sigma_{c_2}$  that achieves exact control refinement for  $\Sigma_{c_1}$  as

$$\Sigma_{c_2} := (\Sigma_1 \times_{w_1} \Sigma_{c_1}) \times_{w_1} \Sigma_{\mathcal{F}},$$

where  $w_1 = (u_1, x_1)$  and where  $\Sigma_{\mathcal{F}} := (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\mathcal{F}})$  is a dynamical system taking values in the combined signal space with

$$\begin{aligned} \mathfrak{B}_{\mathcal{F}} &:= \{(x_1, u_1, x_2, u_2) \in \mathbb{W} \mid x_{10} = \mathcal{F}_0(x_{20}) \text{ and} \\ &\quad u_2 = \mathcal{F}(x_1, u_1, x_2)\}. \end{aligned}$$

The dynamical system  $\Sigma_{c_2}$  is a well-posed controller for  $\Sigma_2$  with  $\Sigma_2 \times_{w_2} \Sigma_{c_2}$  sharing  $w_2 = (u_2, x_2)$ . Denote with  $\mathfrak{B}_{\Sigma_2 \times \Sigma_{c_2}}$  the behaviour of the controlled system, then due to the construction of  $\Sigma_{\mathcal{F}}$  it follows that  $\mathfrak{B}_{\Sigma_2 \times \Sigma_{c_2}}$  is non-empty and  $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$  such that  $(x_{10}, x_{20})$  has a unique continuation in  $\mathfrak{B}_{\Sigma_2 \times \Sigma_{c_2}}$ . Furthermore it holds that  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_2 \times \Sigma_{c_2}}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_1 \times \Sigma_{c_1}})$ .  $\square$

The design of the controller  $\Sigma_{c_2}$  that achieves exact control refinement for  $\Sigma_{c_1}$  is similar to that in (Tabuada, 2009), which also holds in the behavioural framework.

### 3.3 Driving variable systems

Since it is difficult to control and analyse a DS directly, we develop a transformation to a system representation that is in non-singular DS form and is driven by an auxiliary input. We refer to this non-singular DS as the driving variable (DV) system (Weiland, 1991). We investigate whether the DS and the obtained DV system are bisimilar and behaviourally equivalent. Let us first introduce with a simple example the apparent non-determinism or anti-causality in the DS. Later-on, we show the connections between a DS and its related DV system.

**Example 2.** Consider the DS with dynamics  $(E, A, B, C)$  defined as

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 0.2 \\ 0.5 \end{bmatrix}^T, \quad (10)$$

and  $x(t) = [x_1(t) \ x_2(t) \ x_3(t)]^T$ . In this case, the input  $u(t) = -x_3(t)$  is constrained by the third state component. Now the state trajectories of (10) can be found as follows:

- for a given input sequence  $u : \mathbb{N} \rightarrow \mathbb{U}$ , we have  $x_2(t) = -u(t) - u(t+1)$ , and thus we can use this anti-causal relation of the DS to find the corresponding state trajectories;
- alternatively, we can allow the next state  $x_2(t+1)$  to be freely chosen, and for arbitrary state  $x_2(t)$ , the equations (10) impose constraints on the input sequence that is, therefore, no longer free as  $u(t) = -x_3(t)$ .

We embrace the latter, non-deterministic interpretation of the DS.

This non-determinism can be characterised by introducing an auxiliary driving input of a so-called DV system. We

reorganise the state evolution of (1). For simplicity we omit the time index in  $x(t)$  and  $u(t)$  and denote  $x(t+1)$  as  $x^+$

$$M \begin{bmatrix} x^+ \\ u \end{bmatrix} = Ax, \quad (11)$$

where  $M = [E \ -B]$ . For any  $x$ , we notice that the pairs  $(u, x^+)$  are non-unique due to the non-determinism related to  $x^+$ . If  $M$  has full row rank, then it has a right inverse. This always holds when the DS is reachable (cf. Definition 2-1.1 (Dai, 1989)). In that case we can characterise the non-determinism as follows. Let  $M^+$  be a right inverse of  $M$  such that  $MM^+ = I$  and  $N$  be a matrix such that  $\text{im } N = \ker M$  and  $N^T N = I$ . Then all pairs  $(u, x^+)$  that are compatible with state  $x$  in (11) are parametrised as

$$\begin{bmatrix} x^+ \\ u \end{bmatrix} = M^+ Ax + Ns, \quad (12)$$

where  $s$  is a free variable. We now claim that all transitions  $(x, u, x^+)$  in (12) for some variable  $s$  satisfy (11). To see this, multiply  $M$  on both sides of (12) to regain (11). Now assume that there exists a tuple  $(x, u, x^+)$  satisfying (11) that does not satisfy (12). Then there exists an  $s$  and a vector  $z \neq 0$  that is not an element of the kernel of  $M$  and such that the right side of (12) becomes  $M^+ Ax + Ns + z$ . Multiplying again with  $M$ , we infer that there is an additional non-zero term  $Mz$  and that (11) cannot hold. In conclusion any transition of (11) is also a transition of (12) and vice versa.

*Example 3. [Example 2: cont'd] For the DS of Example 2, the related DV system  $\Sigma_{DV}$  is developed as*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} s(t) \\ u(t) &= [0 \ 0 \ -1] x(t) \\ y(t) &= [0 \ 0.2 \ 0.5] x(t). \end{aligned} \quad (13)$$

*As indicated by (13), the input  $u(t)$  is a function of the state trajectory. The non-determinism of  $x_2(t+1)$  is characterised by  $-s(t)$  for which the auxiliary input  $s$  can be freely selected.*

Let us now formalise the notion of a driving variable representation. We associate a driving variable representation with any given DS (1) by defining a tuple  $(A_d, B_d, C_u, D_u, C)$  and setting

$$\begin{bmatrix} A_d \\ C_u \end{bmatrix} = M^+ A, \quad \begin{bmatrix} B_d \\ D_u \end{bmatrix} = N, \quad (14)$$

where  $N \in \mathbb{R}^{(n+p) \times p}$  has orthonormal columns, that is  $N^T N = I$ . For any given DS, this tuple defines the *driving variable system*  $\Sigma_{DV} = (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\Sigma_{DV}})$ , which maintains the same set of initial states  $\mathbb{X}_0$  and has dynamics

$$\begin{aligned} x(t+1) &= A_d x(t) + B_d s(t) \\ u(t) &= C_u x(t) + D_u s(t) \\ y(t) &= C x(t), \end{aligned} \quad (15)$$

thereby yielding the initialised behaviour

$$\mathfrak{B}_{\Sigma_{DV}} := \{w \in \mathbb{W}^{\mathbb{N}} \mid w = (u, x, y), \exists s \in \mathbb{S}^{\mathbb{N}} \text{ s.t. (15) and } x_0 \in \mathbb{X}_0\}.$$

Next, we propose the following assumption for DS, which will be used in the sequel to develop our main results.

*Assumption 1.* The given DS  $\Sigma$  is a dynamical system with dynamics  $(E, A, B, C)$  such that  $M = [E \ -B]$  has full row rank.

The relationship between a DS and its related DV system is characterised as follows.

*Theorem 9.* Let the DS  $\Sigma$  be given as in (1) satisfying Assumption 1 and let  $\Sigma_{DV} = (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\Sigma_{DV}})$  be defined as in (15). Then

- (1)  $\Sigma$  and  $\Sigma_{DV}$  are bisimilar, that is,  $\Sigma \cong \Sigma_{DV}$ ,
- (2)  $\Sigma$  and  $\Sigma_{DV}$  have equal behaviour, i.e.,  $\mathfrak{B}_{\Sigma_{DV}} = \mathfrak{B}_{\Sigma}$ ,
- (3)  $\Sigma$  and  $\Sigma_{DV}$  have equal output behaviour, that is,  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV}})$ .

**Proof.** For the first statement (1), we define the diagonal relation as  $\mathcal{I} := \{(x, x) \mid x \in \mathbb{X}\}$ . Then  $\mathcal{I}$  is a bisimulation relation between  $\Sigma$  and  $\Sigma_{DV}$ , because by construction their state evolutions can be matched, hence stay in  $\mathcal{I}$ ; and they share the same output map. In addition, since they have the same set of initial states it follows that  $\Sigma \cong \Sigma_{DV}$ .

The second part (2) follows immediately from the derivation of  $\Sigma_{DV}$ , because by construction all the transitions in  $\Sigma$  can be matched by those of  $\Sigma_{DV}$  and vice versa, in addition, they have the same output map. Hence, they share the same signal space  $(\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$  and we can conclude that  $\Sigma$  and  $\Sigma_{DV}$  have equal behaviour.

Additionally, we have that (2) implies (3); via Proposition 6 also (1) implies (3).  $\square$

### 3.4 Main result: exact control refinement for DS

Based on the results developed in the previous subsections, we now derive the solution to the exact control refinement problem in Problem 1. More precisely, subject to the assumption that there exists a simulation relation  $\mathcal{R}$  from  $\Sigma_a$  to  $\Sigma$ , for which in addition holds that  $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$  s.t.  $(x_{a0}, x_0) \in \mathcal{R}$ , we show that for any  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$ , there exists a controller  $\Sigma_c$  for  $\Sigma$  that refines  $\Sigma_{c_a}$  such that  $\Sigma_c \in \mathfrak{C}(\Sigma)$  and  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}})$ .

In the case of Assumption 1, we construct DV systems  $\Sigma_{DV}$  and  $\Sigma_{DV_a}$  for the respective DS systems  $\Sigma$  and  $\Sigma_a$  as a first step. For these systems, we develop the following results on exact control refinement:

- i) The exact control refinement for the DV systems:  
 $\forall \Sigma_{DV_a}^c \in \mathfrak{C}(\Sigma_{DV_a}), \exists \Sigma_{DV}^c \in \mathfrak{C}(\Sigma_{DV}), \text{ s.t.}$   
 $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV} \times \Sigma_{DV}^c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c});$
- ii) The exact control refinement from  $\Sigma_a$  to  $\Sigma_{DV_a}$ :  
 $\forall \Sigma_{c_a} \in \mathfrak{C}(\Sigma_a), \exists \Sigma_{DV_a}^c \in \mathfrak{C}(\Sigma_{DV_a}), \text{ s.t.}$   
 $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c});$
- iii) The exact control refinement from  $\Sigma_{DV}$  to  $\Sigma$ :  
 $\forall \Sigma_{DV}^c \in \mathfrak{C}(\Sigma_{DV}), \exists \Sigma_c \in \mathfrak{C}(\Sigma), \text{ s.t.}$   
 $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV} \times \Sigma_{DV}^c}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}).$

It will be shown that the combination of the elements i)–iii) also implies the construction of the exact control refinement for the concrete and abstract DS.

*i) Exact control refinement for the DV systems.* From Theorem 9, we know that  $\Sigma \cong \Sigma_{DV}$  and  $\Sigma_a \cong \Sigma_{DV_a}$  with respective diagonal relations  $\mathcal{I} := \{(x, x) \mid x \in \mathbb{X}\}$  and  $\mathcal{I}_a := \{(x_a, x_a) \mid x_a \in \mathbb{X}_a\}$ . Hence as depicted in Fig. 2 and based on the transitivity of simulation relations, we also derive that  $\mathcal{R}$  is a simulation relation from  $\Sigma_{DV_a}$  to  $\Sigma_{DV}$ .

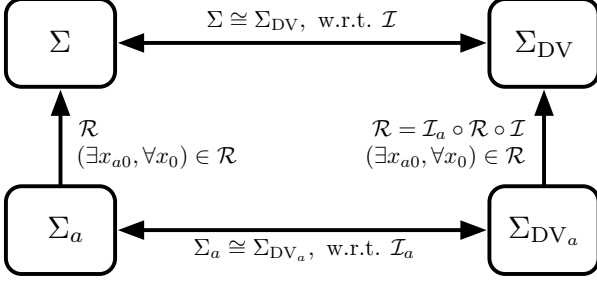


Fig. 2. Connection between DS and DV systems for the exact control refinement.

Since the DV systems  $\Sigma_{DV}$  and  $\Sigma_{DV_a}$  share the same initial states as the respective DS  $\Sigma$  and  $\Sigma_a$ , it also holds that  $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$  s.t.  $(x_{a0}, x_0) \in \mathcal{R}$ . According to Theorem 8, we know that we can do exact control refinement, that is, we have shown

$$\forall \Sigma_{DV_a}^c \in \mathfrak{C}(\Sigma_{DV_a}), \exists \Sigma_{DV}^c \in \mathfrak{C}(\Sigma_{DV}), \text{ s.t. } \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV} \times \Sigma_{DV_a}^c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c}).$$

ii) *Exact control refinement from  $\Sigma_a$  to  $\Sigma_{DV_a}$ .* Denote with  $\Sigma_{DV_a}$  the abstract DV system related to  $\Sigma_a$ , with dynamics  $(A_{da}, B_{da}, C_{ua}, D_{ua}, C_a)$  and initialised with  $\mathbb{X}_{a0}$ . We first derive the static function  $\mathcal{S}_a$  mapping transitions of  $\Sigma_a$  to the auxiliary input  $s_a$  of  $\Sigma_{DV_a}$ . From the definition of DV systems, we can also derive the transitions of  $\Sigma_{DV_a}$  indexed with  $a$ , which is similar to the derivation of (12).

$$\begin{bmatrix} x_a^+ \\ u_a \end{bmatrix} = M_a^+ A_a x_a + N_a s_a. \quad (16)$$

Multiplying  $N_a^T$  on both sides of (16),  $\mathcal{S}_a$  is derived as

$$\mathcal{S}_a : s_a = \mathcal{S}_a(x_a^+, u_a, x_a) = N_a^T \begin{bmatrix} x_a^+ \\ u_a \end{bmatrix} - N_a^T M_a^+ A_a x_a. \quad (17)$$

$\mathcal{S}_a$  maps the state evolutions of  $\Sigma_a \times_{w_a} \Sigma_{c_a}$  to the auxiliary input  $s_a$  for  $\Sigma_{DV_a}$ , where  $w_a = (u_a, x_a)$ . Now, we consider the exact control refinement from the abstract DS to the abstract DV system.

**Theorem 10.** Let  $\Sigma_a$  be the abstract DS with dynamics  $(E_a, A_a, B_a, C_a)$  satisfying the condition of Assumption 1 and let  $\Sigma_{DV_a}$  be its related DV system with dynamics  $(A_{da}, B_{da}, C_{ua}, D_{ua}, C_a)$  such that both systems are initialised with  $\mathbb{X}_{a0}$ . Then, for any  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$ , there exists a controller  $\Sigma_{DV_a}^c \in \mathfrak{C}(\Sigma_{DV_a})$  that is an exact control refinement for  $\Sigma_{c_a}$  as defined in Definition 4 with

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c}).$$

**Proof.** Denote with  $x_a$  and  $x_a^d$  the state variables of  $\Sigma_a$  and  $\Sigma_{DV_a}$ , respectively. Next, we construct the controller  $\Sigma_{DV_a}^c$  that achieves exact control refinement for  $\Sigma_{c_a}$  as

$$\Sigma_{DV_a}^c := (\Sigma_a \times_{w_a} \Sigma_{c_a}) \times_{w_a} \Sigma_{\mathcal{S}_a},$$

where  $w_a = (u_a, x_a)$  and where  $\Sigma_{\mathcal{S}_a} := (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\mathcal{S}_a})$  is a dynamical system with

$$\mathfrak{B}_{\mathcal{S}_a} := \{(x_a, u_a, x_a^d, s_a) \in \mathbb{W} | x_{a0} = x_{a0}^d \text{ and } s_a = \mathcal{S}_a(x_a^+, u_a, x_a)\}.$$

The dynamical system  $\Sigma_{DV_a}^c$  is a well-posed controller for  $\Sigma_{DV_a}$  with  $\Sigma_{DV_a} \times_{w_a^d} \Sigma_{DV_a}^c$  sharing  $w_a^d = (s_a, x_a^d)$ . Denote with  $\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c}$  the behaviour of the controlled system.

By construction, we know that the set of the behaviour is non-empty and there is a unique continuation for any  $x_{a0}^d \in \mathbb{X}_{a0}$ . Further based on the construction of  $\Sigma_{\mathcal{S}_a}$ , the behaviour is such that  $x_a^d(t) = x_a(t), \forall t \in \mathbb{N}$ . Additionally, since  $\Sigma_a$  and  $\Sigma_{DV_a}$  share the same set of initial states  $\mathbb{X}_{a0}$ , it holds that  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV_a} \times \Sigma_{DV_a}^c})$ .  $\square$

The proof is actually constructive in the design of the controller  $\Sigma_{DV_a}^c$  that achieves exact control refinement for  $\Sigma_{c_a}$ .

iii) *Exact control refinement from  $\Sigma_{DV}$  to  $\Sigma$ .* Now, we consider the exact control refinement from  $\Sigma_{DV}$  to  $\Sigma$ . Suppose we are given a well-posed controller  $\Sigma_{DV}^c$  for  $\Sigma_{DV}$ , which shares the free variable  $s$  and the state variable  $x$  with  $\Sigma_{DV}$ . We want to design a well-posed controller for  $\Sigma$  over  $w = (u, x)$ , for which we consider the dynamical system  $\Sigma_c = (\mathbb{N}, \mathbb{W}, \mathfrak{B})$  over the signal space  $\mathbb{W} = \mathbb{U} \times \mathbb{X} \times \mathbb{S}$ , the behaviour of which can be defined by

$$\begin{aligned} B_d^T x(t+1) &= B_d^T A_d x(t) + B_d^T B_d s(t) \\ u(t) &= C_u x(t) + D_u s(t). \end{aligned} \quad (18)$$

Then the dynamics of the interconnected system  $\Sigma \times_w \Sigma_c$  as a function of  $x$  and  $s$  is derived as

$$\begin{bmatrix} E \\ B_d^T \end{bmatrix} x(t+1) = \begin{bmatrix} A + BC_u \\ B_d^T A_d \end{bmatrix} x(t) + \begin{bmatrix} BD_u \\ B_d^T B_d \end{bmatrix} s(t). \quad (19)$$

Note that  $A + BC_u = EA_d$  and  $BD_u = EB_d$  by multiplying  $M = [E \ -B]$  on the left-hand side of the two equations in (14). Therefore, (19) is simplified to

$$\begin{bmatrix} E \\ B_d^T \end{bmatrix} x(t+1) = \begin{bmatrix} E \\ B_d^T \end{bmatrix} A_d x(t) + \begin{bmatrix} E \\ B_d^T \end{bmatrix} B_d s(t). \quad (20)$$

Furthermore  $[E^T \ B_d^T]^T$  has full column rank because the matrix  $[M^T \ N]^T$  is square and has full rank. Hence  $[E^T \ B_d^T]^T$  has a left inverse and the dynamics of  $\Sigma \times_w \Sigma_c$  in (20) can be simplified as

$$x(t+1) = A_d x(t) + B_d s(t),$$

which is exactly the same as the state evolutions of  $\Sigma_{DV}$  as shown in (15). Next we construct  $\Sigma_c := \Sigma_c \times_{w^d} \Sigma_{DV}^c$  with  $w^d = (s, x^d)$  and it is a well-posed controller for  $\Sigma$ . This allows us to state the following theorem regarding the control refinement from  $\Sigma_{DV}$  to  $\Sigma$ .

**Theorem 11.** Let  $\Sigma$  be the concrete DS with dynamics  $(E, A, B, C)$  satisfying Assumption 1 and let  $\Sigma_{DV}$  be its related DV system with dynamics  $(A_d, B_d, C_u, D_u, C)$  such that both systems are initialised with  $\mathbb{X}_0$ . Then, for any  $\Sigma_{DV}^c \in \mathfrak{C}(\Sigma_{DV})$ , there exists a controller  $\Sigma_c \in \mathfrak{C}(\Sigma)$  that is an exact control refinement for  $\Sigma_{DV}^c$  as defined in Definition 4 with

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV} \times \Sigma_{DV}^c}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}).$$

**Proof.** Denote with  $x$  and  $x^d$  the state variables of the  $\Sigma$  and  $\Sigma_{DV}$ , respectively. Next, we construct the controller  $\Sigma_c$  that achieves exact control refinement for  $\Sigma_{DV}^c$  as

$$\Sigma_c := \Sigma_c \times_{w^d} \Sigma_{DV}^c,$$

where  $w^d = (s, x^d)$  and the dynamics of  $\Sigma_c$  is defined as (18). Then, we can show that the dynamical system  $\Sigma_c$  is a well-posed controller for  $\Sigma$ . Based on the analysis of (20), it is shown that  $\Sigma \times_w \Sigma_c = \Sigma_{DV}$  with  $w = (u, x)$ , then we

can derive  $\Sigma \times_w \Sigma_c = \Sigma_{DV} \times_{w^d} \Sigma_{DV}^c$ . Therefore, we can conclude  $\Sigma_c \in \mathfrak{C}(\Sigma)$  with  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_{DV} \times \Sigma_{DV}^c}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c})$  immediately follows from  $\Sigma_{DV}^c \in \mathfrak{C}(\Sigma_{DV})$ .  $\square$

*Exact control refinement for descriptor systems.* We can now argue that there exists exact control refinement from  $\Sigma_a$  to  $\Sigma$ , as stated in the following result.

**Theorem 12.** Consider two DS  $\Sigma_a$  (abstract, initialised with  $\mathbb{X}_{a0}$ ) and  $\Sigma$  (concrete, initialised with  $\mathbb{X}_0$ ) satisfying Assumption 1 and let  $\mathcal{R}$  be a simulation relation from  $\Sigma_a$  to  $\Sigma$ , for which in addition holds that  $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$  s.t.  $(x_{a0}, x_0) \in \mathcal{R}$ . Then, for any  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$ , there exists a controller  $\Sigma_c \in \mathfrak{C}(\Sigma)$  such that

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}}).$$

**Proof.** Based on Assumption 1, we first construct  $\Sigma_{DV}$  and  $\Sigma_{DV_a}$ . Then to prove this we need to construct the exact control refinement. This can be done based on the subsequent control refinements given in Theorem 10, Theorem 8 and Theorem 11.  $\square$

Theorem 12 claims the existence of such controller  $\Sigma_c$  that achieves exact control refinement for  $\Sigma_{c_a}$ . More precisely, we have shown in the proof that the refined controller  $\Sigma_c$  is constructive, which provides the solution to Problem 1.

To elucidate how such an exact control refinement is constructed, we consider the following example.

*Example 4. [Example 2,3: cont'd]* Consider the DS of Example 2 and its related DV system (cf. Example 3) such that both systems are initialised with  $\mathbb{X}_0 = \{x_0 \mid x_0 \in [-1, 1]^3 \subset \mathbb{R}^3\}$ . According to Silverman-Ho algorithm (Dai, 1989), we can select an abstract DS  $\Sigma_a = (E_a, A_a, B_a, C_a)$  that is the minimal realisation of  $\Sigma$  and is initialised with  $\mathbb{X}_{a0} = \mathbb{R}^2$ , in addition

$$E_a = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, A_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B_a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_a = \begin{bmatrix} 0.7 \\ 0.2 \end{bmatrix}^T.$$

Similarly, the related DV system  $\Sigma_{DV_a}$  of  $\Sigma_a$  is given as

$$\begin{aligned} x_a(t+1) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_a(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} s_a(t) \\ u_a(t) &= [-1 \ 0] x_a(t) \\ y_a(t) &= [0.7 \ 0.2] x_a(t). \end{aligned} \quad (21)$$

Subsequently,

$$\mathcal{R} := \{(x_a, x) \mid x_a = \mathcal{H}x, x_a \in \mathbb{X}_a, x \in \mathbb{X}\}$$

is a simulation relation from  $\Sigma_a$  to  $\Sigma$  with

$$\mathcal{H} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}.$$

This can be proved through verifying the two properties of Definition 5. In addition, the condition  $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$  s.t.  $(x_{a0}, x_0) \in \mathcal{R}$  holds. According to Theorem 12, we can refine any  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$  to attain a well-posed controller  $\Sigma_c$  for  $\Sigma$  that solves Problem 1 as follows: Define  $\Sigma_{c_a} \in \mathfrak{C}(\Sigma_a)$  with dynamics as

$$\begin{bmatrix} 1 & 1 \end{bmatrix} x_a(t+1) = [0.5 \ 0.5] x_a(t) + u_a(t).$$

The controlled system  $\Sigma_a \times_{w_a} \Sigma_{c_a}$  is derived as

$$\begin{aligned} x_a(t+1) &= \begin{bmatrix} 0 \\ -0.5 & -0.5 \end{bmatrix} x_a(t) \\ y_a(t) &= [0.7 \ 0.2] x_a(t), \end{aligned}$$

with  $w_a = (u_a, x_a)$  and  $u_a(t) = [-1 \ 0] x_a(t)$ . Then  $\Sigma_a \times_{w_a} \Sigma_{c_a}$  is stable. According to Theorem 10, we derive the map  $\mathcal{S}_a$  for  $\Sigma_{DV_a}$  as  $s_a(t) = [0 \ -1] x_a(t+1) = [0.5 \ 0.5] x_a(t)$ . Next, the related interface from  $\Sigma_{DV_a}$  to  $\Sigma_{DV}$  is developed

as  $s(t) = s_a(t) - [0 \ 1 \ -1] x(t)$ . According to Theorem 11, we derive the well-posed controller  $\Sigma_c$  as

$$\begin{aligned} [0 \ -1 \ 0] x(t+1) &= [0 \ -1 \ 1] x(t) + [0.5 \ 0.5] x_a(t) \\ u(t) &= [0 \ 0 \ -1] x(t), \end{aligned}$$

and the interconnected system  $\Sigma \times_w \Sigma_c$  with  $w = (u, x)$ , is derived as

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} -0.5 & -0.5 \\ 0 & 0 \end{bmatrix} x_a(t) \\ y(t) &= [0 \ 0.2 \ 0.5] x(t). \end{aligned}$$

Since  $(x_a, x) \in \mathcal{R}$ , that is  $x_a = \mathcal{H}x$ ,  $\Sigma \times_w \Sigma_c$  can be simplified by replacing  $x_a(t)$ :

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 0.5 & 1 \\ 0 & 1 & -1 \end{bmatrix} x(t) \\ y(t) &= [0 \ 0.2 \ 0.5] x(t). \end{aligned}$$

Finally,  $\Sigma_c \in \mathfrak{C}(\Sigma)$  and  $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma \times \Sigma_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\Sigma_a \times \Sigma_{c_a}})$  are achieved.

## 4. CONCLUSION

In this paper, we have developed a control refinement procedure for discrete-time descriptor systems that is largely based on the behavioural theory of dynamical systems and the theory of simulation relations among dynamical systems. Our main results provide complete solutions of the control refinement problem for this class of discrete-time systems.

The exact control refinement that has been developed in this work also opens the possibilities for *approximate* control refinement notions, to be coupled with approximate similarity relations: these promise to leverage general model reduction techniques and to provide more freedom for the analysis and control of descriptor systems.

The future research includes a comparison of the control refinement approach for descriptor systems to results in perturbation theory, as well as control refinement for nonlinear descriptor systems.

## REFERENCES

- Abadir, K.M. and Magnus, J.R. (2005). *Matrix algebra*. Cambridge University Press.
- Antoulas, A.C. (2005). *Approximation of large-scale dynamical systems*. SIAM.
- Cao, X., Saltik, M., and Weiland, S. (2015). Hankel model reduction for descriptor systems. In *2015 54th IEEE CDC*, 4668–4673.
- Dai, L. (1989). *Singular control systems*. Springer-Verlag New York, Inc.
- Fainekos, G.E., Girard, A., and Pappas, G.J. (2007). Hierarchical synthesis of hybrid controllers from temporal logic specifications. In *International Workshop on HSCC*, 203–216.
- Girard, A. and Pappas, G.J. (2009). Hierarchical control system design using approximate simulation. *Automatica*, 45(2), 566–571.
- Girard, A. and Pappas, G.J. (2011). Approximate bisimulation: A bridge between computer science and control theory. *European Journal of Control*, 17(5), 568–578.
- Kloetzer, M. and Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1), 287–297.

- Kunkel, P. and Mehrmann, V.L. (2006). *Differential-algebraic equations: analysis and numerical solution*. European Mathematical Society.
- Megawati, N.Y. and Van der Schaft, A. (2015). Bisimulation equivalence of DAE systems. *arXiv:1512.04689*.
- Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.
- Van der Schaft, A. (2004). Equivalence of dynamical systems by bisimulation. *IEEE transactions on automatic control*, 49(12), 2160–2172.
- Weiland, S. (1991). *Theory of approximation and disturbance attenuation for linear systems*. University of Groningen.
- Willems, J.C. and Polderman, J.W. (2013). *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media.